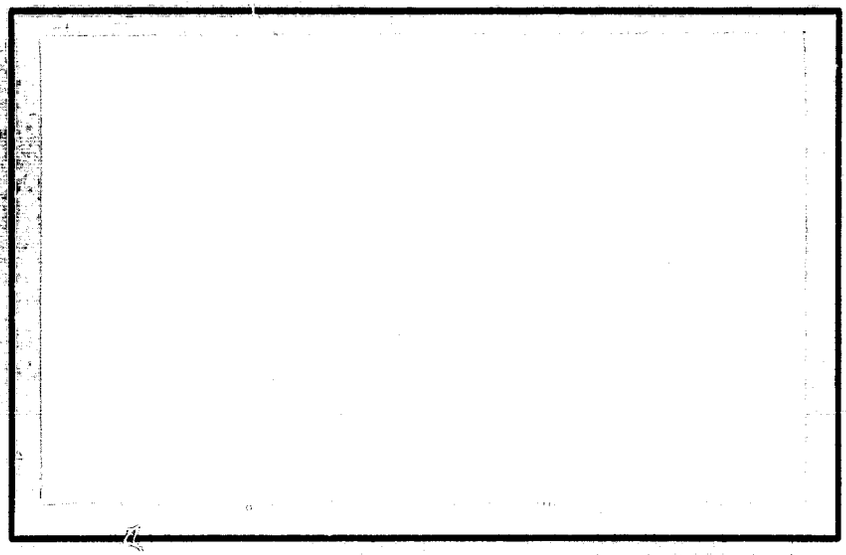
Appendix C

UNIVERSITY OF COLORADO



(NASA-CR-163421) SOFTWARE DEVELOPMENT TOOLS: A BIBLIOGRAPHY, APPENDIX C. (Colorado Univ. at Boulder.) 13 p HC A02/MF A01

N80-30064

CSCL 09B

Unclas

G3/61 28336

DEPARTMENT OF COMPUTER SCIENCE

Technical Report

SOFTWARE DEVELOPMENT TOOLS:
A BIBLIOGRAPHY

William E. Riddle
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309

CU-CS-183-80 June, 1980

This work was supported in part, by grant NSG 1638
from NASA Langley Research Center.

SOFTWARE DEVELOPMENT TOOLS: A BIBLIOGRAPHY

William E. Riddle
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309

Abstract: We give a bibliography on tools which 1) help software developers perform some development task (such as text manipulation, testing, etc.), and 2) which would not necessarily be found as part of a computing facility. The bibliography comes from a relatively random sampling of the literature and is not complete. But it is indicative of the nature and range of tools currently being prepared or currently available.

This work was supported in part, by grant NSG 1638 from NASA Langley Research Center.

The following citations indicate articles which discuss tools of use in the development of software systems. In compiling this list, we did not include references on:

1. notational tools, that is, languages for the abstract description of software systems,
2. cognitive tools, that is, methods or principles intended to help development practitioners order their tasks and decisions,
3. development environments, that is, collections of tools.

Thus we have focused on augmentive tools, that is, tools which help developers in the performance of some development task such as text manipulation, testing, etc.

We have not included references on "traditional" tools such as compilers, linking loaders, sort routines, file systems, etc. Thus we have focused on tools which one would not *pro forma* expect to be available under a computing facility. Further, we have included references to techniques which have been developed but not as yet implemented as complete augmentive tools.

This bibliography is not complete. It was augmented somewhat by using the bibliographies prepared as part of term projects by students in the author's graduate-level software engineering course.

It is, however, felt that its randomness is perhaps a strength in that a quick perusal will indicate the nature and breadth of tools currently available.

M.D. Abrams. A comparative sampling of the systems for producing computer-drawn flowcharts. *Proc. 1968 ACM National Computer Conf.*, (1968), pp. 743-750.

A. Amschler, L. Gmeiner and V. Voger. SADAT: Static and dynamic analysis and testing. *Digest of Workshop on Software Testing and Test Documentation*, Fort Lauderdale, Florida, (December 1978).

S. Andler. Synchronization primitives and the verification of concurrent programs. *Proc. Second International Symp. on Operating Systems*, IRIA, Le Chesnay, France, (October 1978).

G. Ashby, L. Salmonson and R. Heilman. Design of an interactive debugger for Fortran: MANTIS. *Software: Practice and Experience, 3*, (1973), 65-74.

B. Baker. An algorithm for structuring flowgraphs. *J. ACM, 24,* 1 (January 1977), 98-120.

L.A. Belady, J.A. Cavanagh and C.J. Evangelisti. Greenprint: A graphical representation for structured programs. RC7763, IBM T.J. Watson Research Center, Yorktown Heights, New York, (July 1979).

J. Bicevskis, J. Borzovs, E.F. Miller, V. Straujums and A. Zarins. SMOTL: A system to construct samples for data processing program debugging. *IEEE Trans. on Software Engineering, SE-5,* (January 1979), 60-66.

L.E. Bonanni and C.A. Salemi. Source code control system user's guide. Bell Lab. Tech Pub.

R.A. Bowen and F.J. Drasch. DBUG: A tool for program development. *Proc. of the Software Quality Assurance Workshop,* (November 1978).

R.S. Boyer, B. Elspas and K.N. Levitt. SELECT: A formal system for testing and debugging programs by symbolic execution. *SIGPLAN Notices, 10,* 6 (June 1975), 221-227.

G. Bristow. Deadlock detection in HAL/S codes. HAL/S Memo #5, Dept. of Computer Sci., Univ. of Colorado at Boulder, (May 1978).

G. Bristow, C. Drey, B. Edwards and W. Riddle. Anomaly detection in concurrent programs. *Proc. Fourth International Conf. on Software Engineering,* Munich, Germany, (September 1979), pp. 265-273.

J.R. Brown and R.H. Hoffman. A survey of techniques and automated tools. TRW-SS-72-03, TRW Inc., Redondo Beach, California, (May 1972).

J.R. Brown, A.J. DeSalvio, D.E. Heine and J.G. Purdy. Automated software quality assurance. In Hetzel (ed.), *Program Test Methods,* Prentice-Hall, Englewood Cliffs, New Jersey, (1973).

J.R. Brown. Why Tools? *Proc. Computer Sci. and Statistics: Eighth Annual Symp. on the Interface,* (February 1975), pp. 310-312.

J.C. Browne and David B. Johnson. FAST: A second generation program analysis system. *Proc. Third International Conf. on Software Engineering,* Atlanta, Georgia, (May 1978), pp. 142-148.

T.E. Cheatham, G.H. Holloway and J.A. Townley. Symbolic evaluation and the analysis of programs. *IEEE Trans. on Software Engineering, SE-5,* 4 (July 1979), 402-417.

T.S. Chow. Testing software design modelled by finite state machines. *Proc. Compsac 77,* Chicago, Illinois (November 1977), pp. 58-64.

L.A. Clarke. A system to generate test data and symbolically execute programs. *IEEE Trans. on Software Engineering, SE-2,* 3 (September 1976), 215-222.

D. Comer. MOUSE4: An improved implementation of the RATFOR preprocessor. *Software: Practice and Experience, 8,* (January-February 1978), 35-40.

M.N. Condict, R.L. Marcus and A. Mickel. Spruce: Partial program formatter. Univ. of Minnesota Computing Center, (1978).

L.W. Cooprider. The representation of families of software systems. Ph.D. Thesis, Dept. of Computer Sci., Carnegie-Mellon Univ., Pittsburgh, Pennsylvania, (1979).

J.A. Darringer and J.C. King. Applications of symbolic execution to program testing. *Computer, 11,* 4 (April 1978), 51-60.

J.A. Darringer. The use of symbolic execution in program testing. In *Software Testing,* Infotech International Ltd., Maidenhead, England, (1979), pp. 65-84.

G. deBalbine. MTR — A tool for displaying the global structure of software systems. *Proc. 1978 National Computer Conf.,* Anaheim, California, (June 1978), pp. 571-580.

E. Denert. The project library: A tool for software development. SOFTLAB GmbH, Munich, West Germany, (1979).

P. Deutsch. A flexible measurement tool for software systems. *Proc. IFIP Congress 71,* Ljubljana, Yugoslavia, (August 1971), pp. TA-3-7-TA-3-12.

F.M. Djorup and E.T. Irons. A CRT editing system. *Comm. ACM, 15,* 11 (January 1972).

T.A. Dolotta, et al. The LEAP load and test driver. *Proc. Second International Conf. on Software Engineering,* San Francisco, California, (October 1976), 182-186.

J.J. Donovan. Tools and philosophy for software education. *Comm. ACM, 19,* 8 (August 1976), 430-436.

I.B. Elliott. SPTRAN: A Fortran compatible structured programming language converter. *Proc. Symp. on Computer Software Engineering,* (1976).

B. Elspas, K.N. Levitt, R.J. Waldinger and A. Waksman. An assessment of techniques for proving program correctness. *Computing Surveys, 4,* 2 (June 1972), 97-147.

R.E. Fairley. An experimental program-testing facility. *IEEE Trans. on Software Engineering, SE-1,* 4 (December 1975), 350-357.

R.E. Fairley. Tutorial: Static analysis and dynamic testing of computer software. *Computer,* (April 1978), 14.

R.E. Fairley. ALADDIN: Assembly language assertion driven debugging interpreter. *IEEE Trans on Software Engineering, SE-5,* 4 (July 1979).

R.E. Fairley. Software validation and pre-implementation issues. In Riddle and Fairley (eds.), *Software Development Tools,* Springer Verlag, Heidelberg, Germany, (1980).

S.I. Feldman. Make: A program for maintaining computer programs. Bell Lab. Tech. Pub.

D. Ferrari and M. Liu. A general-purpose software measurement tool. *Software: Practice and Experience, 5,* (April-June 1975), 181-192.

A. Fitzsimmons and T. Love. A review and evaluation of software science. *Computing Surveys, 10,* 1 (March 1978), 3-18.

R.J. Fleischer and R.W. Spitler. SIMON: A project management system for software development. *Proc. Symp. on Computer Software Engineering,* (1976).

L.D. Fosdick and L.J. Osterweil. Data flow analysis in software reliability. *Computing Surveys, 8,* 3 (September 1976), 305-330.

L.D. Fosdick and L.J. Osterweil. DAVE: A validation, error detection and documentation system for FORTRAN programs. *Software: Practice and Experience, 6,* (October-December 1976), 473-486.

C.W. Fraser. A generalized text editor. *Comm. ACM, 23,* 3 (March 1980), 154-158.

S.L. Gaede. Tools for research in computer workload characterization and modelling. Bell Lab., Naperville, Illinois, (1980).

L. Gallenson, et al. PRIM User's Manual. ISI/TM-75-1, Information Sci. Inst., Univ. of Southern California, Marina del Rey, California, (1975).

T.J. Gilkey, J.R. White and T.L. Booth. Performance analysis as a software design tool. *Proc. Compsac 77,* Chicago, Illinois, (November 1977).

R. Gimson. The application of a data model to the support of Fortran programs. PMM/79, Computing Lab., The Univ. of Newcastle upon Tyne, England, (1978).

A.L. Glasser. The evaluation of a source code control system. *Proc. of the Software Quality Assurance Workshop,* (November 1978).

J. Goguen, J. Tardo, N. Williamson and M. Zainfir. A practical method for testing algebraic specifications. Computer Sci. Dept., Univ. of California, Los Angeles, California.

D.I. Good, R.L. London and W.W. Blesdoe. An interactive program verification system. *IEEE Trans. on Software Engineering, SE-1,* (March 1975), 59-67.

J.B. Goodenough and D.T. Ross. The effect of software structure on software reliability, modifiability, reusability and efficiency: A preliminary analysis. 1007-2, Softech, Waltham, Massachusetts, (July 1973).

J.B. Goodenough and S.L. Gerhart. Towards a theory of test data selection. *IEEE Trans on Software Engineering, SE-1,* 2 (June 1975), 156-173.

A.N. Habermann. A software development control system. Dept. of Computer Science, Carnegie-Mellon Univ., Pittsburgh, Pennsylvania, (1979).

A.N. Habermann. Tools for software system construction. In Riddle and Fairley (eds.), *Software Development Tools,* Springer Verlag, Heidelberg, Germany, (1980).

R. Hamlet, J. Gannon, M. Ardis and P. McMullin. Testing data abstractions through their abstractions. TR-761, Dept. of Computer Sci., Univ. of Maryland, College Park, Maryland, (May 1979).

S.L. Hantler and J.C. King. An introduction to proving the correctness of programs. *Computing Surveys, 8,* 3 (September 1976).

D.F. Hare. A structure editor for the Gypsy verification environment. ICSCA-CMP-16, Certifiable Minicomputer Project, Univ. of Texas at Austin, Texas, (July 1979).

R.D. Hartwick. Test planning: *Proc. AFIPS National Computer Conf., 46,* AFIPS Press, Montvale, New Jersey, (1977), pp. 285-294.

P.G. Hebalkar. Contour-forming, a mechanism for procedure autpmation in the office of the future. RI 2492, IBM Research Lab., San Jose, California, (March 1979).

P. Henderson. Finite state modelling in program development. *SIGPLAN Notices, 10,* 6 (June 1975), 221-227.

P. Henderson, R. Gimson, G.D. Pratten and R.A. Snowdon. The maintenance of software with multiple versions. In *Structured System Design,* Infotech International Ltd., Maidenhead, England, (1979).

P. Henderson. Relational data models for describing system structure. Computing Lab., Univ. of Newcastle upon Tyne, England.

W.C. Hetzel (ed.). *Program Test Methods.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, (1972).

C.A. Heuermann, G.J. Myers and J.H. Winterton. Automated test and verification. *IBM Tech. Disclosure Bulletin, 17,* 7 (1974), 2030-2035.

C.R. Hollander. Decompilation of object programs. TR-54, Stanford Electronics Lab., Stanford Univ., Stanford, California, (January 1973).

W.E. Howden and L.G. Stucki. Final report: Methodology for effective test case selection, Phase II. MDCG5800, McDonnell Douglas Corp., (April 1975).

W.E. Howden. Experiments with a symbolic execution system. *Proc. AFIPS National Computer Cont., 45,* (1976), pp. 899-908.

W.E. Howden. DISSECT: A symbolic evaluation and program testing system. *IEEE Trans. on Software Engineering, SE-4,* 1 (January 1978), 70-73.

W.E. Howden. An evaluation of the effectiveness of symbolic testing. *Software: Practice and Experience, 8,* 4 (July-August 1978), 381-397.

W.E. Howden. Functional program testing. *IEEE Trans. on Software Engineering, SE-6,* (1980).

J.C. Huang. Detection of data flow anomaly through program instrumentation. *IEEE Trans. on Software Engineering, SE-5,* 3, (1979) 226-236.

C.E. Hughes and C.P. Pfleeger. ASSIST-V: An environment simulator for IBM 360 Systems software development. *IEEE Trans. on Software Engineering, SE-4,* 6 (November 1978).

J. Humbrecht. DEMDAB: A design and maintenance data base system. M.S. Thesis, Dept. of Computer Sci., Univ. of Colorado at Boulder, Boulder, Colorado, (to appear 1980).

J. Hurras, H. Legard and R.L. Marcus. Pretty: A Pascal pretty printing program. Univ. of Minnesota Computing Center, (1978).

D. Ingalls. The smalltalk-76 programming system. *Proc. Fifth Annual ACM Symp. on Principles of Programming Languages,* Tucson, Arizona, (January 1978), pp. 9-16.

F.S. Ingrassia. Combating the 90% complete syndrome. *Datamation, 24,* 1 (January 1978), 171-176.

J.H. Jacobs and T.I. Dillon. Interactive saturn flight program simulator. *IBM Systems J., 9,* 2 (1970), 145-158.

S.C. Johnson. Lint, a C program checker. Bell lab. Tech. Pub.

B.W. Kernighan. RATFOR: A preprocessor for a rational Fortran. *Software: Practice and Experience, 9,* (January 1979), 73-86.

B.W. Kernighan and P.J. Plauger. *Software Tools.* Addison Wesley, Reading, Massachusetts, (1976).

B.W. Kernighan, M.E. Lesk and J.F. Ossanna. Unix time-sharing system: Document preparation. *Bell Systems Tech. J., 57,* 6 (1978).

M.L. Kersten and A.I. Wasserman. The architecture of the PLAIN data base handler. Submitted for publication, (1979).

M.L. Kersten, A.I. Wasserman and R.P. van de Reit. TROLL: A data base interface and testing tool. Submitted for publication, (1980).

J.C. King. A new approach to program testing. *SIGPLAN Notices, 10,* 6 (June 1975), 228-233.

J.C. King. Symbolic execution and program testing. *Comm. ACM, 19,* 7 (July 1976), 385-394.

D.B. Knudsen, et al. A modification request control system. *Proc. Second International Conf. on Software Engineering,* San Francisco, California, (October 1976), 187-192.

K.W. Krause, R.W. Smith and M.A. Goodwin. Optimal software test planning through automated network analysis. *Record 1973 IEEE Symp. on Computer Software Reliability,* IEEE, New York, (1973), pp. 18-22.

H.E. Kulsrud. Helper: An interactive extensible debugging system. *Proc. Second ACM Symp. on Operating Systems Principles,* Princeton, New Jersey, (1969), pp. 105-111.

V.R. Lesser and J.C. Wileden. Issues in the design of tools for distributed software systems development. In Riddle and Fairley (eds.), *Software Development Tools,* Springer Verlag, Heidelberg, Germany, (1980).

R.L. London. A view of program verification. *SIGPLAN Notices, 10,* 6 (June 1975), 534-545.

R.L. London and L. Robinson. The role of verification tools and techniques. In Riddle and Fairley (eds.), *Software Development Tools,* Springer Verlag, Heidelberg, Germany, (1980).

W.J. Long. A program writer. MIT/LCS/TR-87, Lab. for Computer Sci., Massachusetts Inst. of Tech., Cambridge, Massachusetts, (November 1977).

D.C. Luckham, S.M. German, F.W. Henke, R.A. Karp, P.W. Milne. The Stanford Pascal verifier user manual. Stan-CS-79-731, SVG-11, Computer Sci. Dept., Stanford, California (March 1979).

K.D. Mack. CLEAN: A computer program for reformatting Fortran programs. Air Force Aero Propulsion Lab., Wright-Patterson Air Force Base, (May 1979).

Z. Manna. *Mathematical Theory of Computation.* McGraw-Hill, New York, 1974.

J.R. Mashey and D.W. Smith. Documentation tools and techniques. *Proc. Second International Conf. on Software Engineering*, San Francisco, California, (October 1976), 177-181.

B.E. Meyer and H.J. Schneider. Tools for information system design and realization. In Schneider (ed.), *Formal Models and Practical Tools for Information Systems Design*, North-Holland Pub. Co., Amsterdam, The Netherlands, (1979).

E.F. Miller. Methodology for comprehensive software testing. RADC-TR-75-161, General Research Corp., Santa Barbara, California, (1975).

E.F. Miller. Program testing tools: A survey. *Proc. MIDCON Conf.*, Chicago, Illinois, (November 1977).

E.F. Miller. Program testing: An overview for managers. In *Software testing*, Infotech International Ltd., Maidenhead, England, (1979), pp. 187-200.

P.R. Mohilner, Pretty printing Pascal programs. *SIGPLAN Notices, 13*, 7 (July 1978), 34-40.

G.J. Myers. *The Art of Software Testing*. John Wiley and Sons, New York, New York, (1979).

I. Nassi and B. Schneiderman. Flowchart techniques for structured programming. *SIGPLAN Notices, 8*, 8 (August 1973), 12-26.

I. Nassi. A critical look at the process of tool development: Industrial perspective. In Riddle and Fairley (eds.), *Software Development Tools*, Springer-Verlag, Heidelberg, Germany (1980).

N. Ng. A graphical editor for programming using structured charts. RJ2344, IBM Research Lab., San Jose, California, (September 1978).

G. Nutt. Evaluation net simulation system reference manual. CU-CS-042-74, Dept. of Computer Sci., Univ. of Colorado at Boulder, Boulder Colorado, (April 1974).

L.J. Osterweil. A proposal for an integrated testing system for computer programs. CU-CS-93-76, Dept. of Computer Sci., Univ. of Colorado at Boulder, Boulder, Colorado, (August 1976).

L.J. Osterweil and L.D. Fosdick. Some experience with DAVE, a Fortran program analyzer. *Proc. AFIPS National Computer Conf., 45*, (1976), pp. 909-916.

L.J. Osterweil. A methodology for testing computer programs. *Proc. AIAA Conf. on Computers in Aerospace*, Los Angeles, California, (November 1977), pp. 52-62.

D.J. Panzl. Automatic software test drivers. *Computer, 11*, 4 (1978), 44-50.

R. Parnes, H.C. Hench and K. Zinn. A brief user guide to Confer II for computer-based conferencing. Center for Research on Learning and Teaching, Univ. of Michigan, Ann Arbor, Michigan, (April 1976).

R. Parnes and H. Hench. Primer for Confer II: An introduction to computer-based conferencing. Center for Research on Learning and Teaching, Univ. of Michigan, Ann Arbor, Michigan, (October 1975).

J.L. Peterson. On the formatting of Pascal programs. *SIGPLAN Notices, 12,* 12 (December 1977), 83-86.

R.J. Peterson. TESTER/1: An abstract model for the automatic synthesis of program test case specifications. *Proc. Symp. on Computer Software Engineering,* Polytechnic Press, New York, (1976), pp. 465-484.

J.W. Pomeroy. A guide to programming tools and techniques. *IBM Systems J., 11,* 3 (1972), 340-359.

L. Presser and R. Hug. Portable software. *Computerworld,* (April 14, 1980).

C.V. Ramamoorthy, R.E. Meeker and J. Turner. Design and construction of an automated software evaluation system. *Record of the 1973 IEEE Symp. on Computer Software Reliability,* IEEE, New York, (1973), pp. 28-37.

C.V. Ramamoorthy and S.F. Ho. Fortran automatic code evaluation system (FACES). ERL-M466, Dept. of Electrical and Computer Engineering, Univ. of California, Berkeley, California, (1974).

C.V. Ramamoorthy and S.F. Ho. Testing large software with automated software evaluation system. *IEEE Trans. on Software Engineering, SE-1,* 1 (March 1975), 46-58.

C.V. Ramamoorthy, S.F. Ho and W.T. Chen. On the automated generation of program test data. *IEEE Trans. on Software Engineering, SE-2,* 4 (December 1976), 293.

C.V. Ramamoorthy, S.F. Ho and H.H. So. The role of software tools in a methodology for the development and validation of critical software for nuclear power plants. In Ramamoorthy and Yeh (ed.), *Tutorial: Software Methodology,* IEEE, New York, (1978).

D.J. Reifer. Automated aids for reliable software. *SIGPLAN Notices, 10,* 6 (June 1975), 131-142.

D.J. Reifer and S. Trattner. A glossary of software tools and techniques. *Computer, 10,* 7 (July 1977), 52-59.

W. Riddle. An approach to software system modelling and analysis. *Computer Languages, 4,* (1979), 49-66.

L. Robinson and K. Levitt. Proof techniques for hierarchically structured programs. In Yeh, (ed.), *Current Trends in Programming Methodology, Vol.II,* Prentice Hall Inc., Englewood Cliffs, N.J., (1974).

M.J. Rochking. The source code control system. *IEEE Trans. on Software Engineering, SE-1,* 4 (December 1975), 364-370.

R. Rustin (ed.). *Debugging Techniques in Large Systems.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, (1970).

J.W. Sanguinetti. Performance prediction in an operating system design methodology. RSSM/32 (Ph.D. Thesis), Dept. of Computer and Comm. Sci., Univ. of Michigan, Ann Arbor, Michigan, (May 1977).

J.W. Sanguinetti. A formal technique for analyzing the performance of complex systems. *Proc. CPEUG Conf.*, Boston, Massachusetts, (October 1978).

J.W. Sanguinetti. A technique for integrating simulation and system design. *Proc. Conf. on Simulation, Measurement and Modeling of Computer Systems*, Boulder, Colorado, (August 1979), pp. 163-172.

E. Satterthwaite. Debugging tools for high-level languages. Software: Practice and Experience, (1972).

F. Sayward, R. Demillo and R. Lipton. Program mutations: A new approach to program testing. *Proc. Software Specification and Testing Technology Conf.*, Office of Naval Research, (April 1978).

A. Saxena. The static detection of deadlocks. CU-CS-122-77, Dept. of Computer Sci., Univ. of Colorado at Boulder, Boulder, Colo., (November 1977).

R.S. Scowen. Some aids for program documentation. *Software: Practice and Experience, 7*, (1977), 779-792.

R.L. Sites. Programming tools: Statement counts and procedure tunings. *SIGPLAN Notices, 13*, 12 (December 1978), 98-101.

A.M. Stavely. A survey of feedback aids in software design aid systems. RSSM/60, Dept. of Computer and Comm. Sci., Univ. of Michigan, Ann Arbor, (November 1977).

A.M. Stavely. The membership problem for behaviors of concurrent software systems. RSSM/93 and CSR 153, Computer Sci. Dept., New Mexico Inst. of Mining and Tech., Socorro, New Mexico, (September 1979).

S.A. Stephens and L.L. Tripp. Requirements expression and validation aid. *Proc. Third International Conf. on Software Engineering*, Atlanta, Georgia, (May 1978), pp. 101-108.

L.G. Stucki and G.L. Foshee. New assertion concepts for self-metric software validation. *SIGPLAN Notices, 10*, 6 (June 1975), 59-71.

L.G. Stucki. The use of dynamic assertions to improve software quality. Ph.D. Thesis, School of Engineering, Univ. of California at Los Angeles, California, (June 1976).

L.G. Stucki. New directions in automated tools for improving software quality. In Yeh (ed.), *Current Trends in Programming Methodology, Volume II, Program Validation*, Prentice-Hall, Englewood Cliffs, New Jersey (1977), pp. 80-111.

R.N. Taylor and L.J. Osterweil. A facility for verification, testing and documentation of concurrent process software. *Proc. Compsac 78*, Chicago, (November 1978), pp. 36-41.

D.H. Thompson and S.L. Gerhart. Affirm Reference Library: I, Reference Manual; II, User's Guide; III, Type Library; IV, Annotated Transcripts; V, Collected Papers. Info. Sci. Inst., Univ. of Southern California, Marina del Rey, California, (November 1979).

W. Tichy. Software development control based on module interconnection. *Proc. Fourth International Conf. on Software Engineering*, Munich, Germany, (September 1979), pp. 29-41.

E.C. Van Horn. Software evolution using the SEER data base. Digital Equipment Corp., Maynard, Masachusetts, (June 1978).

U. Voges, L. Gmeiner and A.A. von Mayrhauser. SADAT: An automated testing tool. *IEEE Trans. on Software Engineering, SE-6*, 3 (May 1980), 286-290.

R.R. Willis. DAS: An automated system to support design analysis. *Proc. Third International Conf. on Software Engineering*, Atlanta, Georgia, (May 1978).

G. Wong. Design methodology for computer system modelling tools. In Zeigler (ed.), *Methodology in Systems Modelling and Simulation*, North-Holland Pub. Co., Amsterdam, The Netherlands, (1979).

M.R. Woodward, et al. Experience with path analysis and testing of programs. *IEEE Trans. on Software Engineering, SE-6*, 3 (May 1980), 278-286.

E. Yarwood. Toward program illustration. CSRG-84, Computer Sys. Research Group, Univ. of Toronto, Toronto, Canada, (October 1977).

Fortran Test Procedure Language: Programmer Reference Manual. General Electric Corp., Schenectady, New York, (1977).

RXVP User's Guide. RM-1942; General Research Corp., Santa Barbara, California, (1975).

Proceedings 1975 International Conference on Reliable Software, Los Angeles, California, (April 1975). (Also: *SIGPLAN Notices, 10*, 6 (June 1975).)

Proceedings First International Conference on Software Engineering, Washington, D.C., (1975).

Proceedings Second International Conference on Software Engineering, San Francisco, California, (October 1976).

Proceedings Third International Conference on Software Engineering, Atlanta, Georgia, (May 1978).

Proceedings Fourth International Conference on Software Engineering, Munich, Germany, (September 1979).

IEEE Transaction on Software Engineering, IEEE, New York.

Module Testing System (MTS) Fact Book. Management Systems and Programming Ltd., London, England, (1972).